**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: THỰC TẬP CƠ SỞ**

**MÃ HỌC PHẦN: INT13147**

**ĐỀ TÀI: Xây dựng website tìm kiếm việc làm và tích hợp công nghệ AI hỗ trợ tư vấn**

**Giảng viên hướng dẫn : Ths.Bùi Văn Kiên**

**Nhóm : 10**

**Các sinh viên thực hiện :**

**B22DCCN613 Đỗ Gia Phong**

**B22DCCN693 Ngô Đức Sơn**

**HÀ NỘI 2024**

**MỤC LỤC**

**MỞ ĐẦU**

Trong bối cảnh thị trường lao động ngày càng cạnh tranh, nhiều người gặp khó khăn trong việc **định hướng nghề nghiệp** và **tìm kiếm công việc phù hợp** với năng lực của mình. Không ít ứng viên cảm thấy mơ hồ về ngành nghề phù hợp, không biết kỹ năng của mình có thể đáp ứng được công việc nào, hoặc bối rối trước vô số lựa chọn trên các nền tảng tuyển dụng.

Giải pháp của nhóm em là một **nền tảng tìm việc tích hợp AI**, đóng vai trò như một **chuyên gia tư vấn nghề nghiệp**. Hệ thống sẽ **phân tích kỹ năng**, kinh nghiệm và sở thích của ứng viên, từ đó đề xuất ngành nghề phù hợp cũng như những công việc đang tuyển dụng. Không chỉ dừng lại ở việc tìm việc, AI còn hỗ trợ **định hướng phát triển sự nghiệp**, gợi ý các kỹ năng cần trau dồi để nâng cao cơ hội nghề nghiệp trong tương lai.

Với sự hỗ trợ của AI, quá trình tìm việc không còn là một hành trình mơ hồ mà trở thành một trải nghiệm cá nhân hóa, giúp mỗi ứng viên tìm thấy con đường sự nghiệp phù hợp nhất với mình.

1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG
2. **Giới thiệu hệ thống**

Hệ thống tìm kiếm việc làm tích hợp AI được thiết kế nhằm giúp ứng viên tìm được công việc phù hợp, hỗ trợ nhà tuyển dụng quản lý tuyển dụng hiệu quả và cung cấp công cụ giám sát, phân tích cho quản trị viên. Hệ thống bao gồm **ba vai trò chính: User (Ứng viên), Recruiter (Nhà tuyển dụng) và Admin (Quản trị viên).**

1. **Các thành phần chính của hệ thống**
   1. **Phân hệ User (Người dùng)**

Ứng viên là người dùng chính của hệ thống, có thể :

**Xác thực tài khoản :**

* Đăng ký tài khoản bằng username/password hoặc Google OAuth.
* Đăng nhập, đăng xuất, lấy lại mật khẩu khi quên

**Tìm kiếm và quản lý công việc :**

* Tìm kiếm việc làm theo từ khóa, ngành nghề, địa điểm.
* **Xem chi tiết công việc**, bao gồm mô tả, yêu cầu, lương, địa điểm, công ty đăng tuyển.
* **Ứng tuyển công việc**, theo dõi trạng thái ứng tuyển (**thành công, bị từ chối**).
* Quản lý danh sách **công việc yêu thích**.

**Cá nhân hóa thông tin**

* Chỉnh sửa hồ sơ cá nhân, cập nhật kỹ năng, kinh nghiệm.
* Nhận **tư vấn AI** về ngành nghề phù hợp dựa trên kỹ năng và xu hướng thị trường.
* Nhận **gợi ý công việc** phù hợp thông qua AI.

**Nhận thông báo :**

* Xem danh sách thông báo hệ thống.
* Nhận **thông báo đẩy** về việc tuyển dụng, trạng thái ứng tuyển.
  1. **Phân hệ Recruiter (Nhà tuyển dụng)**

Nhà tuyển dụng có thể:

**Quản lý tin tuyển dụng**:

* Thêm, chỉnh sửa, xóa **công việc tuyển dụng**.
* Xem danh sách công việc đang tuyển.

**Quản lý ứng viên**:

* Xem danh sách ứng viên đã ứng tuyển.
* **Phê duyệt hoặc từ chối** ứng viên.

**Cập nhật thông tin công ty**:

* Chỉnh sửa thông tin công ty hiển thị trên hệ thống.
  1. **Phân hệ Admin (Quản trị viên)**

Quản trị viên chịu trách nhiệm giám sát và vận hành hệ thống:

**Quản lý người dùng:**

* Xem danh sách ứng viên, nhà tuyển dụng.
* Xóa tài khoản vi phạm.

**Quản lý công việc**:

* Giám sát danh sách công việc đang được tuyển dụng.
* Gỡ bỏ các công việc vi phạm chính sách.

**Phân tích & báo cáo**:

* Xem **thống kê theo ngành nghề, số lượng ứng viên, tỉ lệ ứng tuyển thành công**.
* Hỗ trợ **trí tuệ nhân tạo (AI) trong việc phân tích xu hướng tuyển dụng**.

1. **Kiến trúc hệ thống**
   1. **Kiến trúc tổng quan**

Hệ thống được xây dựng theo mô hình **Monolithic** , với các thành phần chính:

**FrontEnd :**

* Sử dụng **HTML, CSS, JS, Bootstrap** để xây dựng giao diện.
* Kết nối với API từ backend để hiển thị dữ liệu theo thời gian thực.

**Backend :**

* Xây dựng bằng **Spring Boot (Java)**
* Cung cấp API RESTful cho frontend.
* Tích hợp AI để phân tích dữ liệu ứng viên.

**Database :**

* **PostgreSQL** để lưu trữ thông tin ứng viên, công việc.

**AI & Machine Learning**:

* Xây dựng mô hình AI giúp tư vấn việc làm, phân tích kỹ năng của ứng viên.
* Gợi ý công việc phù hợp dựa trên lịch sử ứng tuyển và hồ sơ cá nhân.

1. **Quy trình hoạt động của hệ thống**
   1. **Quy trình ứng viên tìm việc**

* Đăng nhập / Đăng ký tài khoản.
* Tìm kiếm công việc theo ngành nghề, địa điểm, mức lương.
* Xem chi tiết công việc và nộp đơn ứng tuyển.
* Theo dõi trạng thái ứng tuyển (đang xét duyệt, thành công, bị từ chối).
* Nhận thông báo khi có cập nhật từ nhà tuyển dụng.
* Nhận tư vấn AI để tìm công việc phù hợp hơn.
  1. **Quy trình nhà tuyển dụng đăng tuyển**
* Đăng nhập tài khoản nhà tuyển dụng.
* Thêm công việc mới, điền thông tin mô tả, yêu cầu, mức lương.
* Quản lý danh sách công việc đang tuyển.
* Xem danh sách ứng viên, phê duyệt hoặc từ chối.
* Cập nhật thông tin công ty để tăng độ tin cậy.
  1. **Quy trình quản trị viên**
* Quản lý người dùng, kiểm tra tài khoản vi phạm.
* Giám sát công việc, loại bỏ công việc không hợp lệ.
* Xem báo cáo phân tích, hỗ trợ cải thiện trải nghiệm người dùng.

1. **Tích hợp AI trong hệ thống**

Trí tuệ nhân tạo (AI) đóng vai trò quan trọng trong việc:

* **Phân tích kỹ năng ứng viên**, xác định ngành nghề phù hợp.
* **Gợi ý công việc tốt nhất** dựa trên hồ sơ cá nhân.
* **Dự đoán xu hướng tuyển dụng**, giúp ứng viên có định hướng phát triển sự nghiệp.

1. MỘT SỐ CÔNG NGHỆ SỬ DỤNG
   * 1. Front End
2. HTML (HyperText Markup Language)

* HTML ngôn ngữ đánh dấu dùng để xây dựng cấu trúc cơ bản của một trang web.
* Chức năng chính :

Định nghĩa các phần tử như: tiêu đề, đoạn văn, hình ảnh, biểu mẫu, nút bấm, bảng, liên kết, v.v.

Là "bộ xương" của giao diện người dùng, giúp trình duyệt hiểu được nội dung cần hiển thị.

1. CSS (Cascading Style Sheets)

* CSS là ngôn ngữ dùng để tạo kiểu dáng, định dạng và làm đẹp cho trang web.
* Chức năng chính :

Quy định màu sắc, phông chữ, bố cục, độ rộng, khoảng cách, hiệu ứng chuyển động,...

Giúp giao diện trở nên trực quan và dễ sử dụng hơn.

1. JS (Javascript)

* Ngôn ngữ lập trình phía client giúp xử lý logic, tương tác, và hành vi động của website.
* Chức năng chính :

Bắt sự kiện người dùng (click, nhập dữ liệu, cuộn trang,...)

Giao tiếp với backend (qua API)

Thay đổi nội dung trang web mà không cần tải lại (AJAX)

1. Bootstrap 5

* Framework CSS & JS phổ biến giúp xây dựng giao diện web nhanh chóng, đẹp và tương thích đa thiết bị (responsive).
* Chức năng chính :

Bộ **component sẵn có**: navbar, button, form, card, modal,...

**Lưới 12 cột (Grid system):** dễ dàng bố trí giao diện responsive.

**Tùy chỉnh nhanh:** thông qua class mà không cần viết CSS nhiều.

* Ưu điểm :

Giảm thời gian phát triển.

Hỗ trợ tốt trên mọi trình duyệt.

Tương thích tốt với thiết bị di động.

* + 1. BackEnd

1. Ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình hướng đối tượng được giới thiệu bởi Sun Microsystems vào năm 1995. Đặc điểm nổi bật của Java là nguyên tắc "Viết một lần, chạy mọi nơi" (WORA), cho phép lập trình viên viết mã mà không cần phải lo lắng về nền tảng cụ thể, vì mã sẽ được biên dịch thành mã bytecode có thể chạy trên mọi thiết bị có Java Virtual Machine (JVM). Ngôn ngữ này cũng tự động quản lý bộ nhớ thông qua cơ chế thu gom rác, giúp giảm thiểu rủi ro khi lập trình viên phải quản lý bộ nhớ. Hơn nữa, Java cung cấp một bộ thư viện phong phú hỗ trợ cho việc phát triển nhiều loại ứng dụng, từ ứng dụng web và di động (như ứng dụng Android) đến các hệ thống lớn phục vụ cho doanh nghiệp. Nhờ vào cộng đồng lập trình viên đông đảo và các tài nguyên phong phú, Java vẫn giữ được vị trí quan trọng trong lĩnh vực công nghệ thông tin, là sự lựa chọn phổ biến cho nhiều dự án phần mềm hiện nay.

1. Thư viện sử dụng

**Spring Data JPA**:

* Spring Data JPA là một phần mở rộng của Spring Data, hỗ trợ tương tác với cơ sở dữ liệu bằng cách cung cấp các API dựa trên JPA (Java Persistence API).
* Nó giúp giảm bớt việc viết mã SQL thủ công, thay vào đó cung cấp các phương thức truy vấn đơn giản với cú pháp rõ ràng và dễ sử dụng.
* Ta có thể tạo các repository để thao tác CRUD (tạo, đọc, cập nhật, xóa) mà không cần viết nhiều mã SQL phức tạp.

**Lombok**:

* Lombok là một thư viện giúp giảm bớt mã boilerplate (mã không cần thiết) trong Java.
* Bằng cách sử dụng các annotation của Lombok như @Getter, @Setter, @Data, và @Builder, ta có thể tự động sinh mã cho các phương thức getter, setter, constructor, builder và nhiều tính năng khác.
* Điều này giúp mã nguồn trở nên ngắn gọn và dễ đọc hơn.

**MapStruct**:

* MapStruct là một thư viện Java dùng để ánh xạ (mapping) giữa các đối tượng DTO và entity một cách tự động.
* Nó giúp chuyển đổi dữ liệu giữa các lớp (classes) một cách nhanh chóng và tránh các lỗi thường gặp trong việc ánh xạ thủ công.
* Bằng cách sử dụng các interface và annotation của MapStruct, bạn có thể định nghĩa các ánh xạ mà không cần viết nhiều mã chuyển đổi.

**Hibernate**:

* Hibernate là một framework ORM (Object-Relational Mapping) phổ biến nhất trong Java, được sử dụng để ánh xạ các lớp Java với các bảng trong cơ sở dữ liệu.
* Nó giúp quản lý các giao dịch, hỗ trợ lazy loading, caching và cung cấp các tính năng như xử lý quan hệ (One-to-Many, Many-to-Many) một cách linh hoạt.
* Hibernate thường được sử dụng cùng với Spring Data JPA, giúp quản lý và truy vấn dữ liệu một cách thuận tiện hơn.

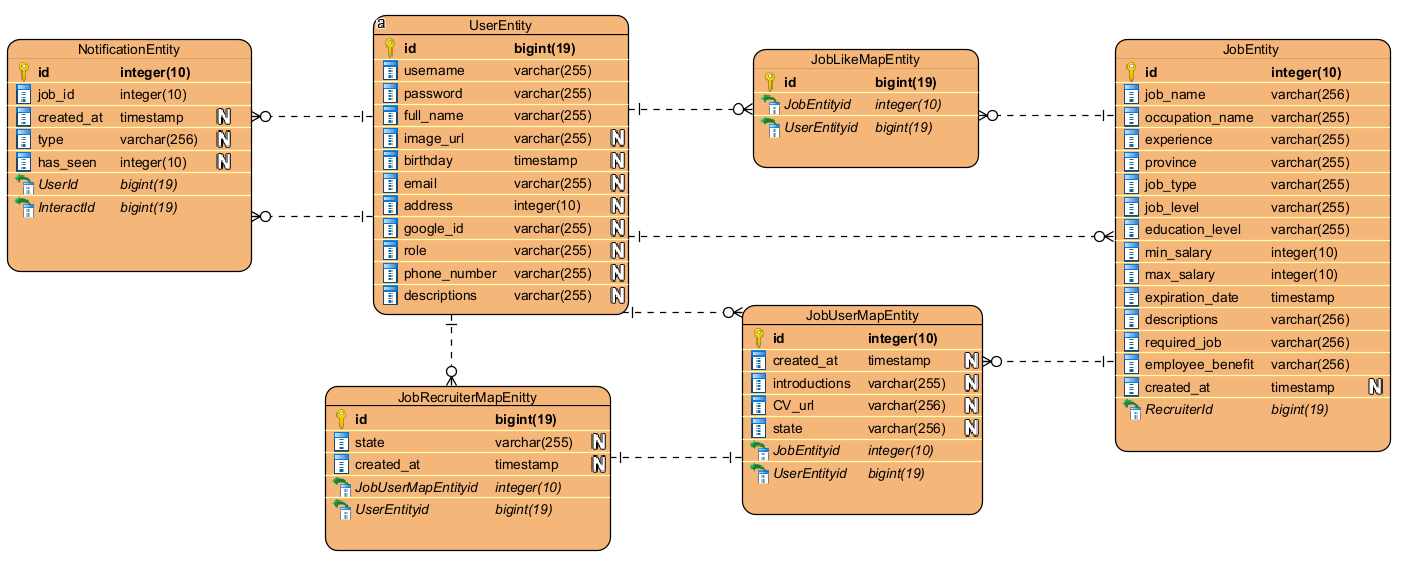
**PostgresSql**

* **PostgresSQL** là một trình điều khiển (driver) JDBC giúp Spring Boot kết nối và tương tác với cơ sở dữ liệu PostgresSQL.
* Khi tích hợp PostgresSQL, ta có thể sử dụng các thư viện như Spring Data JPA và Hibernate để thực hiện các thao tác CRUD, truy vấn dữ liệu, và quản lý giao dịch trên Postgres.
* Driver này giúp chuyển đổi các câu lệnh SQL từ ứng dụng Java thành các lệnh mà Postgres hiểu và thực thi, làm cho việc giao tiếp với cơ sở dữ liệu trở nên liền mạch.

**Cloudinary**:

* Cloudinary là một dịch vụ lưu trữ và quản lý phương tiện truyền thông (media) đám mây, thường được sử dụng để lưu trữ, xử lý và phân phối hình ảnh, video.
* Khi tích hợp Cloudinary, ta có thể tải ảnh và video lên từ ứng dụng của mình và nhận lại URL để truy cập hoặc hiển thị.
* Cloudinary còn cung cấp các API để xử lý ảnh và video, ví dụ như thay đổi kích thước, cắt, nén ảnh, và chuyển đổi định dạng, giúp việc quản lý hình ảnh trong ứng dụng của mình trở nên dễ dàng và hiệu quả.

1. THIẾT KẾ CƠ SỞ DỮ LIỆU



- Hệ thống cơ sở dữ liệu được thiết kế phục vụ cho một nền tảng tuyển dụng trực tuyến, trong đó bao gồm các thực thể chính: **người dùng (UserEntity)**, **công việc (JobEntity)**, **ứng tuyển công việc (JobUserMapEntity)**, **thích công việc (JobLikeMapEntity)**, **thông báo (NotificationEntity)** và **tương tác nhà tuyển dụng (JobRecruiterMapEntity)**. Các thực thể này được liên kết chặt chẽ để hỗ trợ toàn bộ quy trình từ đăng tuyển, tìm kiếm việc làm, ứng tuyển, cho đến quản lý tương tác giữa nhà tuyển dụng và ứng viên.

**- UserEntity** là bảng trung tâm lưu trữ thông tin người dùng, bao gồm cả **ứng viên** và **nhà tuyển dụng, admin**, được phân biệt qua thuộc tính role. Mỗi người dùng ngoài **admin** có thể tạo hồ sơ với thông tin cá nhân như họ tên, ngày sinh, email, số điện thoại, ảnh đại diện…

**- JobEntity** là bảng lưu trữ thông tin về các công việc do người dùng là nhà tuyển dụng đăng tải. Mỗi công việc bao gồm các thông tin như tên công việc, lĩnh vực, loại hình, cấp bậc, mức lương, địa điểm làm việc, yêu cầu về trình độ học vấn và kinh nghiệm. Trường RecruiterId trong bảng này là khóa ngoại tham chiếu đến **UserEntity**, thể hiện người dùng đã đăng công việc đó. Quan hệ giữa **UserEntity** và **JobEntity** là 1-n.

- Mối quan hệ giữa người ứng tuyển và công việc được thể hiện qua bảng **JobUserMapEntity**, đóng vai trò như bảng trung gian lưu thông tin **ứng tuyển**. Một người dùng có thể ứng tuyển nhiều công việc, và mỗi công việc cũng có thể nhận được nhiều lượt ứng tuyển từ các ứng viên. Trong bảng này, các trường như introductions, CV\_url, state, và created\_at ghi nhận nội dung ứng tuyển, link CV, trạng thái xử lý và thời gian ứng tuyển. Bảng này liên kết với **UserEntity** qua UserId và với **JobEntity** qua JobId. Quan hệ giữa người ứng tuyển và công việc ở đây là n-n.

- Ngoài việc ứng tuyển, người dùng còn có thể thể hiện sự quan tâm bằng cách **thích công việc**, được lưu trong bảng **JobLikeMapEntity**. Bảng này thể hiện mối quan hệ nhiều-nhiều giữa người dùng và công việc thông qua khóa ngoại UserId và JobId.

- Để hỗ trợ tương tác giữa hệ thống và người dùng, bảng **NotificationEntity** được sử dụng để lưu các **thông báo** gửi đến người dùng. Thông báo có thể liên quan đến công việc hoặc hành động tương tác (như có người ứng tuyển vào công việc của bạn). Bảng này có các thuộc tính như type (loại thông báo), has\_seen (trạng thái đã xem hay chưa), created\_at (thời gian tạo), và hai khóa ngoại UserId (người nhận thông báo) và InteractId (người tương tác liên quan).

Cuối cùng, bảng **JobRecruiterMapEntity** được dùng để lưu lại trạng thái **tương tác giữa nhà tuyển dụng và ứng viên**, thông qua đơn ứng tuyển (JobUserId) và người dùng là nhà tuyển dụng (RecruiterId). Bảng này giúp nhà tuyển dụng quản lý và cập nhật tiến trình xử lý từng đơn ứng tuyển, chẳng hạn như đang xem xét, chấp nhận, từ chối, hay mời phỏng vấn.

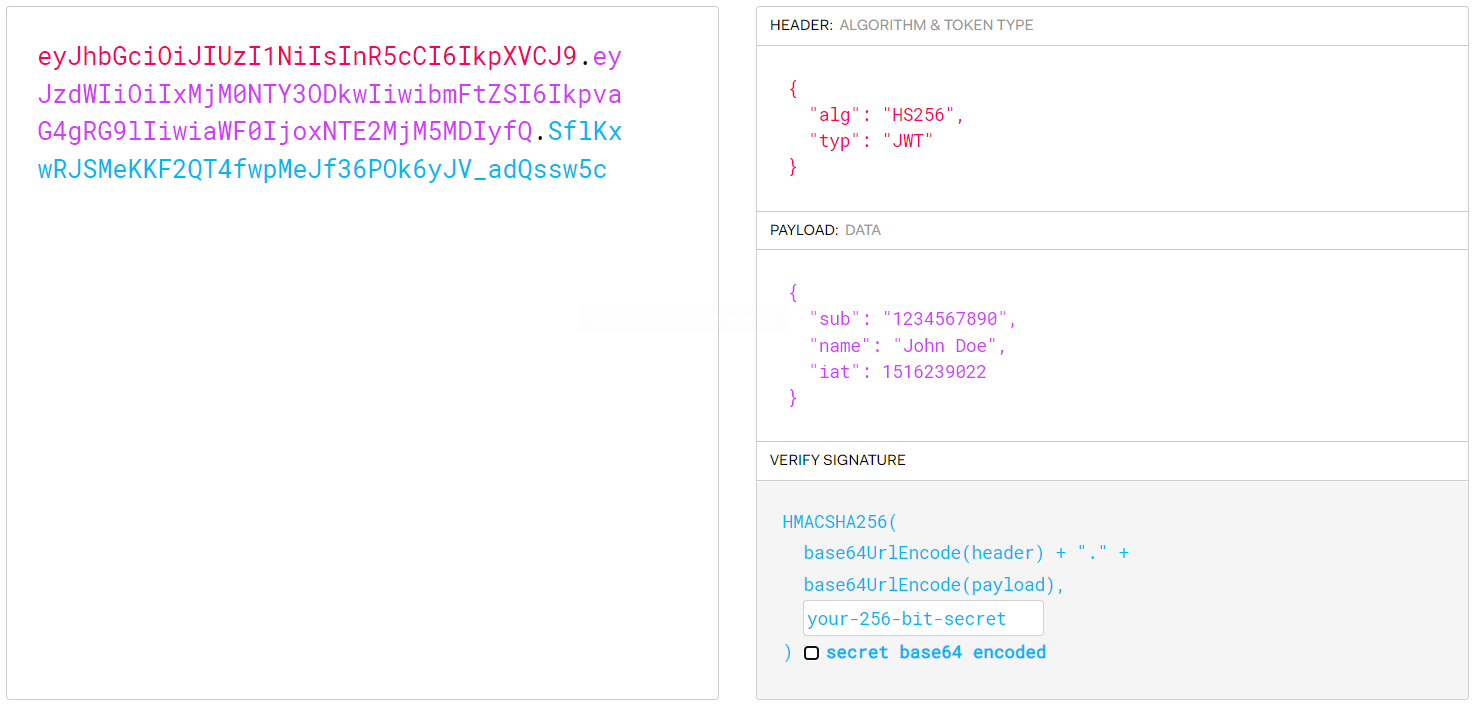
1. CHỨC NĂNG
   * 1. Đăng ký, đăng nhập
2. JSON Web Token

* JSON Web Token là chuỗi mã hóa được sinh ra sau khi người dùng đăng nhập vào ứng dụng, chuỗi mã hóa này được sử dụng để xác minh quyền truy cập vào tài nguyên mà không cần nhập lại username/password nữa.
* Cấu trúc của JWT :
  + Header : gồm 2 phần chính :
    - typ - Loại token (mặc định là JWT - cho biết đây là 1 token JWT)
    - Alg – Là thuật toán dùng để mã hóa.
    - Từ 2 thành phần trên thì header sẽ được mã hóa dựa trên mã hóa

base64url để nhận được đoạn mã bất kì

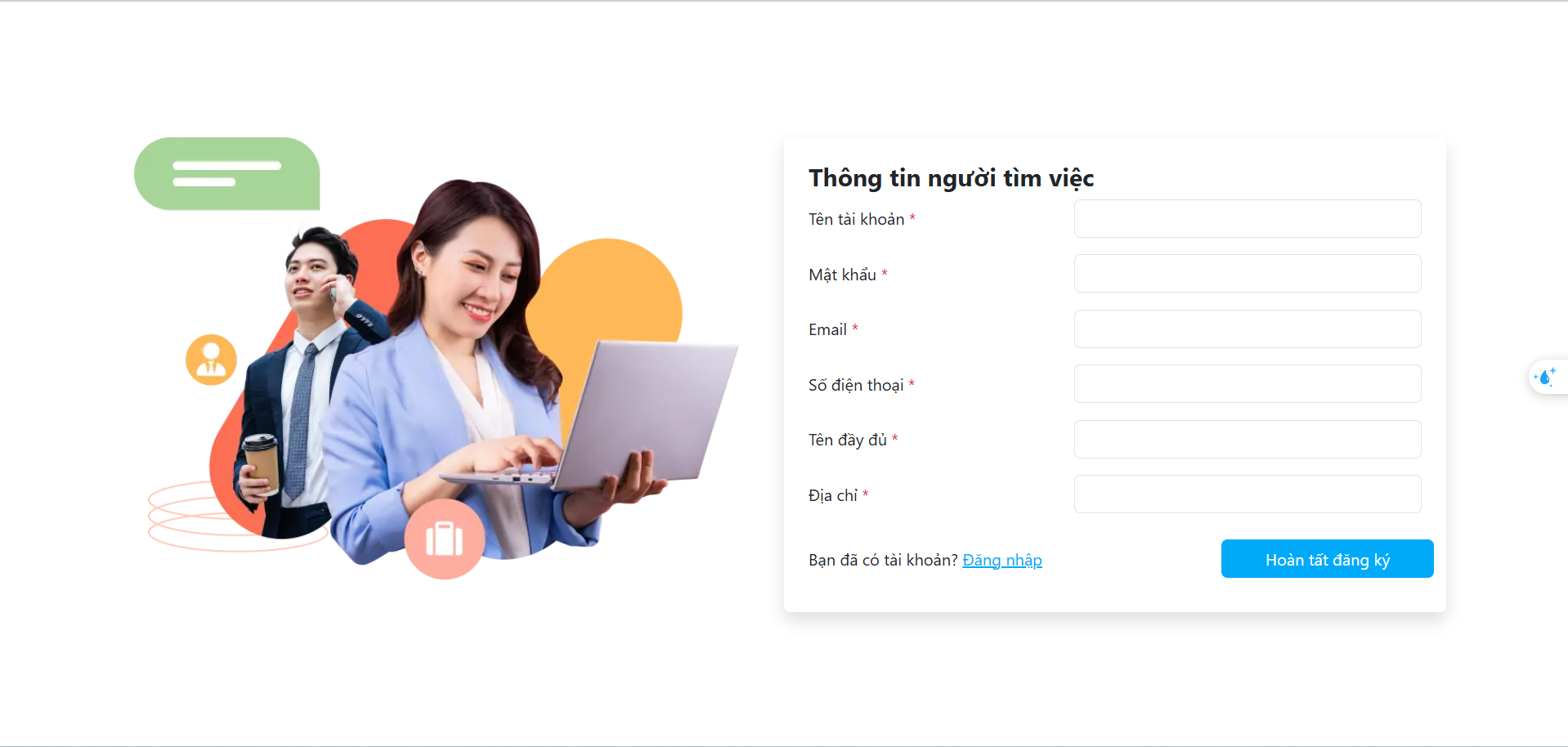
* + Payload : Là nơi chứa các nội dụng thông tin, nó có thể là mô tả 1 thực thể hoặc cũng là cá thông tin bổ sung cho header, VD có thể 1 vài thông tin người dùng hoặc ngày bắt đầu, ngày hết hạn ... Sau đó nó cũng sử dụng mã hóa base64 để nhận được 1 đoạn mã hóa.
* Signature : Là phần kết hợp giữa header và payload đã được mã hóa base64, sau đó nó sẽ kết hợp với 1 cái secretkey và thuật toán mã hóa ở phần header để mã hóa thu được signature cuối cùng. Đây cũng là phần quan trọng nhất để xác định rằng accessToken không bị thay đổi khi truyền tin.
* Sau khi có được 3 thành phần trên, kết hợp chúng và phân tách mỗi phần

bằng dấu chấm.

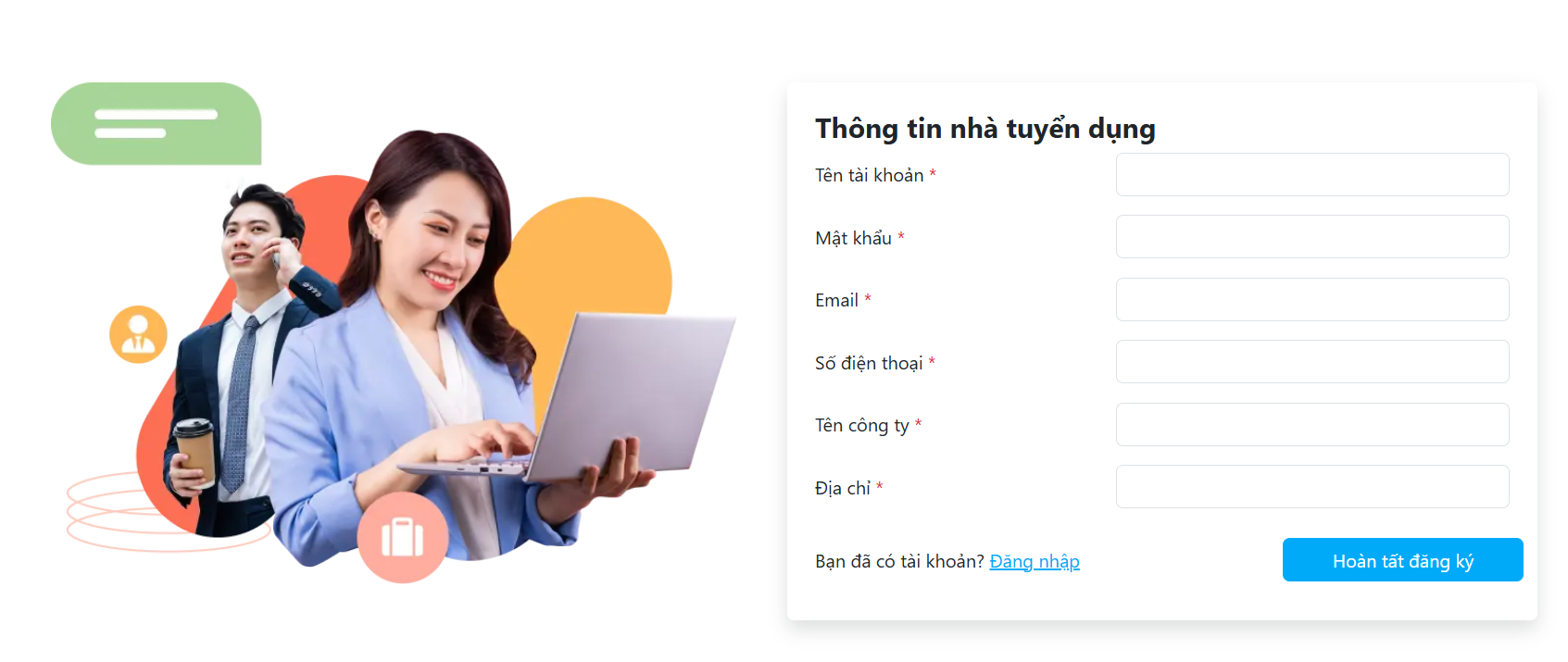


1. Đăng ký

* Ở giao diện đăng ký, tùy vào role mà người đăng ký muốn, sẽ hiển thị ra 2 form đăng ký khác nhau ứng với người tìm kiếm việc làm hoặc nhà tuyển dụng.
* Các thông tin ở form đăng ký sẽ được validate như là tên tài khoản, số điện thoại, email đã tồn tại trong cơ sở dữ liệu hay chưa, nếu đã tồn tại sẽ trả về thông báo cho người dùng.
* Sau khi đăng kí thành công, người dùng sẽ được điều hướng trở lại trang đăng nhập của hệ thống.

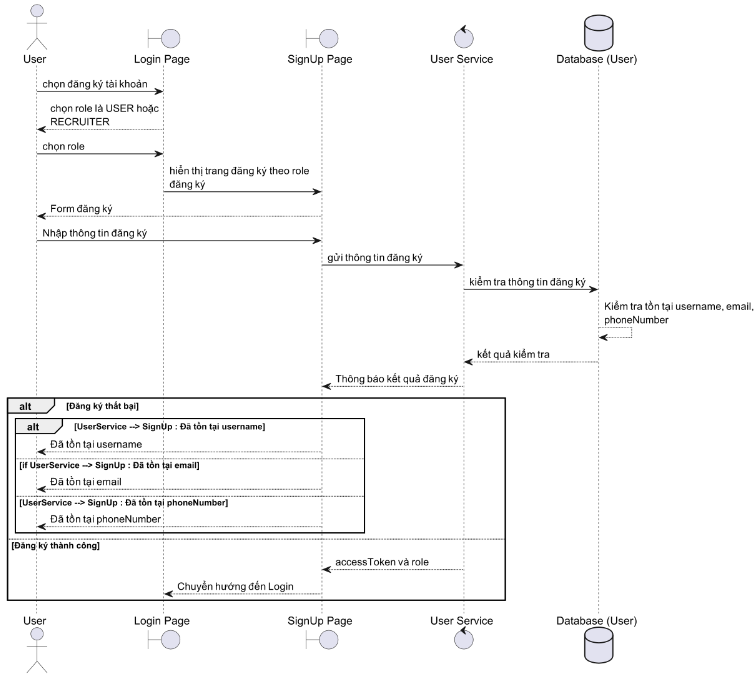


1. Form đăng ký dành cho người tìm việc

****

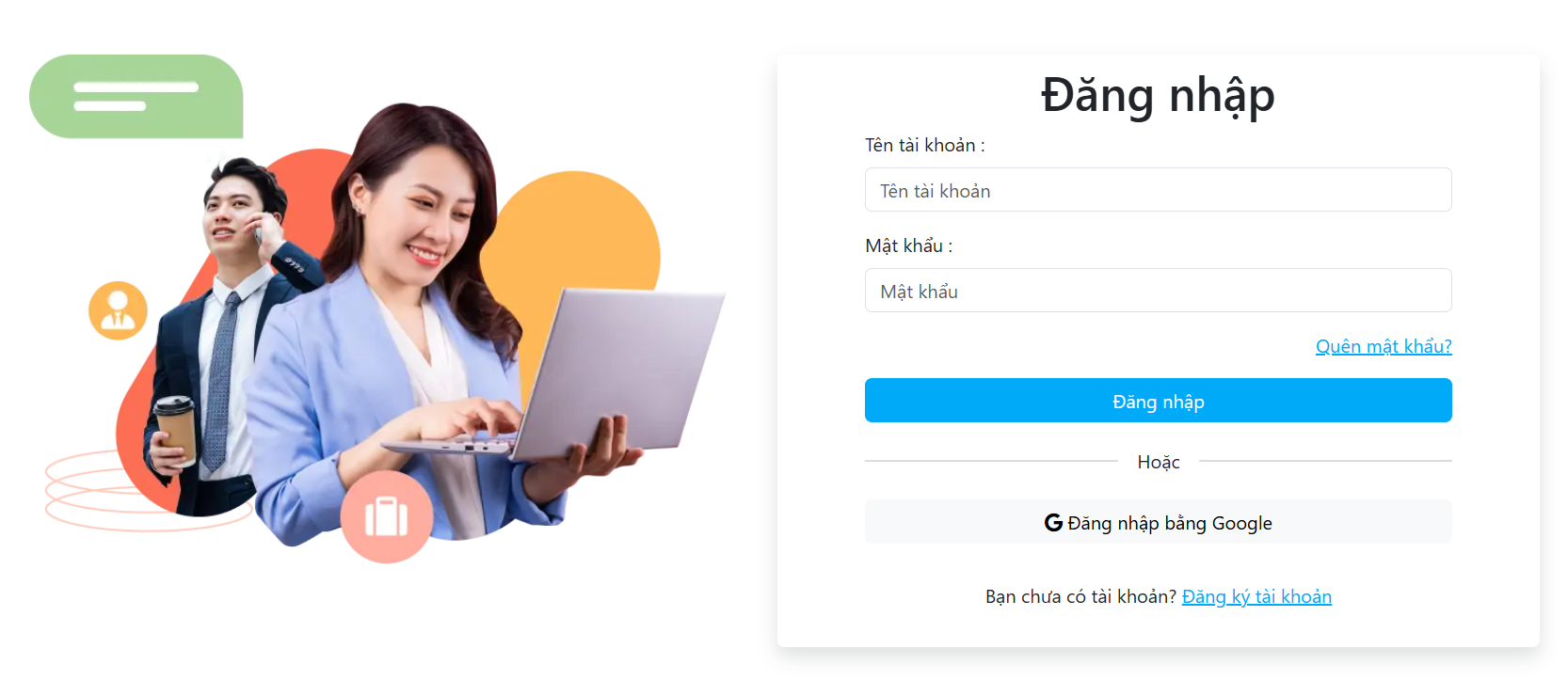
1. Form đăng ký dành cho nhà tuyển dụng

* Tùy thuộc vào role mà người dùng chọn, sẽ cần điền các thông tin khác nhau.
* Dưới đây là sơ đồ luồng hoạt động của việc đăng ký tài khoản :

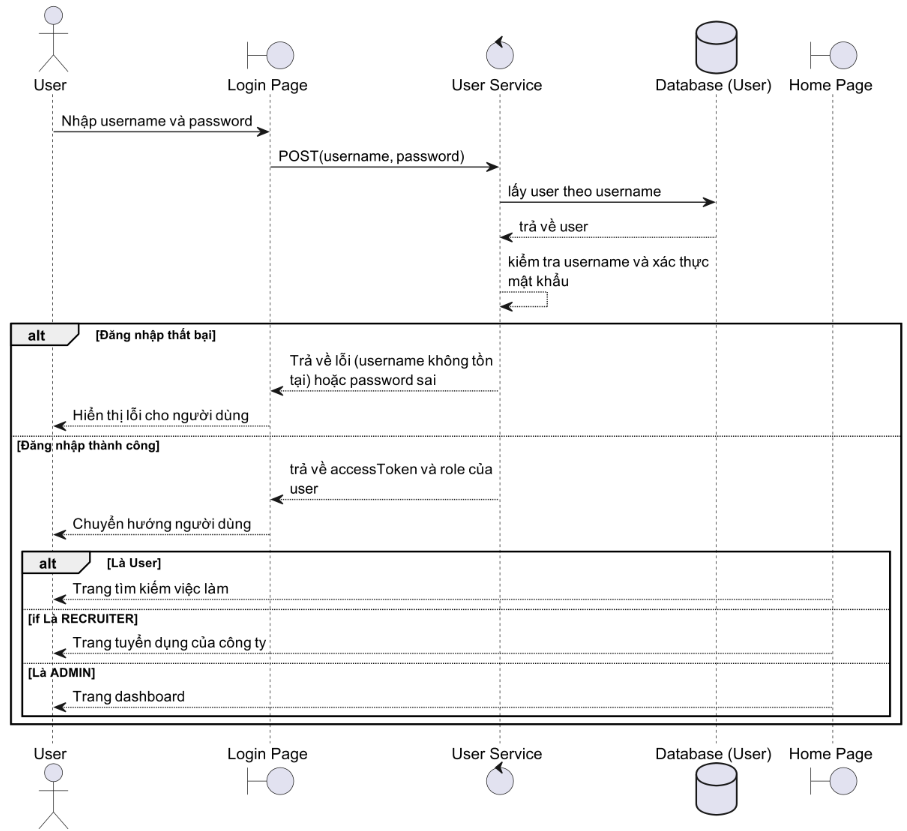


1. Luồng đăng ký tài khoản
2. Đăng nhập sử dụng username/password

* Sau khi đăng ký thành công, người dùng sẽ nhập tài khoản, mật khẩu đã đăng ký từ trước đó để đăng nhập.
* Tại bước đăng nhập, sau khi gửi thông tin đăng nhập, phía server sẽ kiểm tra xem tên tài khoản đã tồn tại hay chưa, đồng thời kiểm tra mật khẩu đã đúng hay chưa.
* Nếu tên tài khoản chưa tồn tại hoặc sai mật khẩu sẽ thông báo trả lại người dùng. Nếu tên tài khoản và mật khẩu đã chính xác, người dùng sẽ được điều hướng đến giao diện ứng với role của người dùng đã đăng ký từ trước đó.
* Đồng thời phía FE cũng sẽ nhận được accessToken và role của người dùng đó. AccessToken được lưu lại và sử dụng cho các yêu cầu sau này của người dùng.
* Dưới đây là hình ảnh giao diện đăng nhập cũng như là luồng hoạt động của chức năng đăng nhập.



1. Giao diện đăng nhập bằng usename/password



1. Luồng hoạt động của chức năng đăng nhập bằng username/password